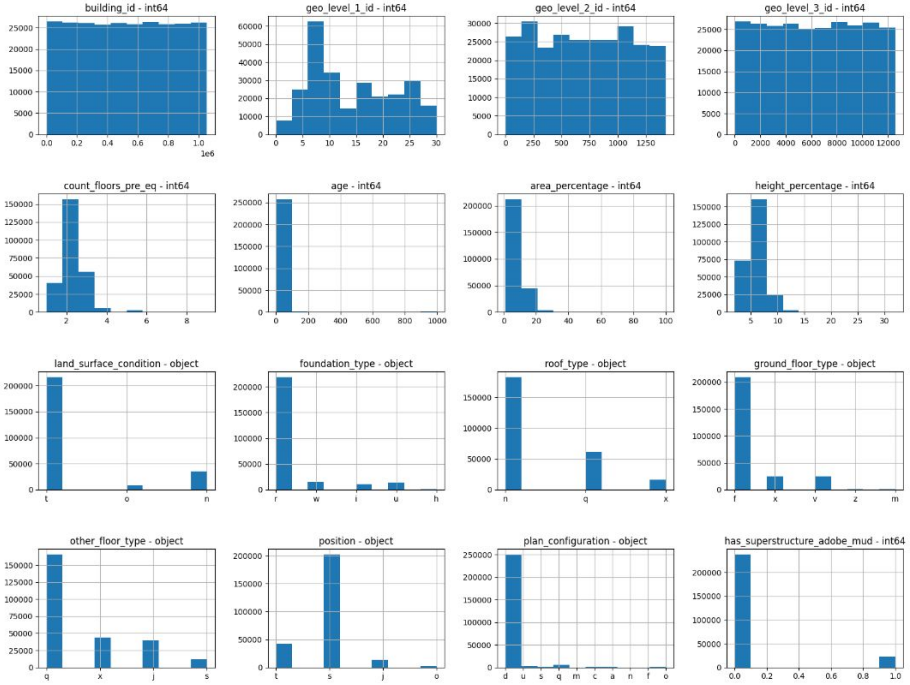
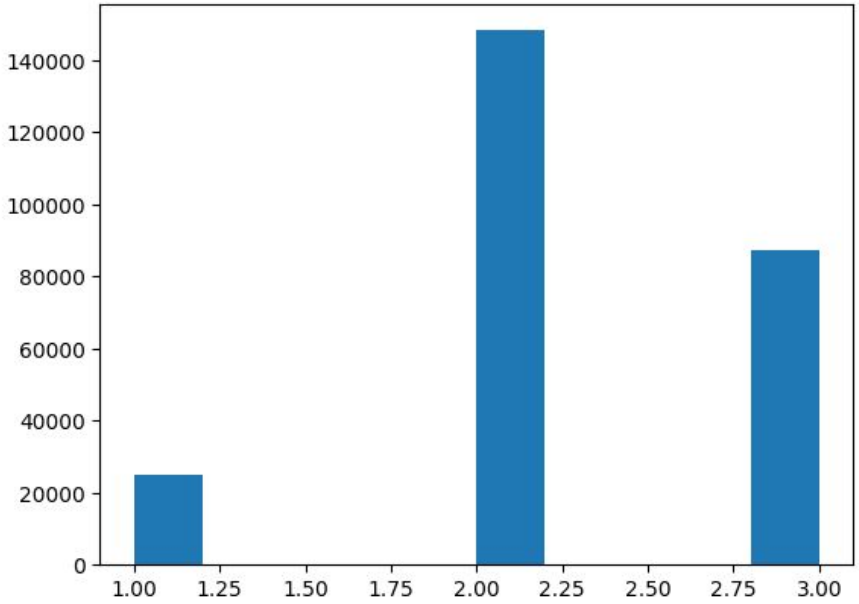
Mini competition

Problem definition



Problem definition



Basic Data exploration

Data cleaning :

Duplicated entries was removed

Outlier removal : using z score

First Basic model

Started with Vanilla Random Forest

- N_estimators: 100
- Max_depth: 6

Performance: → ~.70

Modeling improvement

1 Random Forest with Gridsearch → ~.73

2 XGBoost with GridSearch → ~ .728

3 Stacking with:

```
GaussianNB
RandomForestClassifier
DecisionTreeClassifier
AdaBoostClassifier
LinearDiscriminantAnalysis
GradientBoostingClassifier
LogisticRegression
KNeighborsClassifier
LGBMClassifier
ExtraTreesClassifier
```

→ ~.7325

The following explorations were focussed on improving Stacking

Feature Engineering

```
categorical_feature
```

```
['geo_level_1_id',
 'geo_level_2_id',
 'geo_level_3_id',
 'land_surface_condition',
 'foundation_type',
 'roof_type',
 'ground_floor_type',
 'other_floor_type',
 'position',
 'plan_configuration',
 'has_superstructure_adobe_mud',
 'has_superstructure_mud_mortar_stone',
 'has_superstructure_stone_flag',
 'has_superstructure_cement_mortar_stone',
 'has_superstructure_mud_mortar_brick',
 'has_superstructure_cement_mortar_brick',
 'has_superstructure_timber',
 'has_superstructure_bamboo',
 'has_superstructure_rc_non_engineered',
 'has_superstructure_rc_engineered',
 'has_superstructure_other',
 'legal_ownership_status',
 'has_secondary_use',
 'has_secondary_use_agriculture',
 'has_secondary_use_hotel',
```

```
df["sticking_material"] = df["sticking_material"].map({
    'has_superstructure_mud_mortar_stone': 'mud',
    'has_superstructure_mud_mortar_brick': 'mud',
    'has_superstructure_cement_mortar_stone': 'cement',
    'has_superstructure_cement_mortar_brick': 'cement'
}).fillna("none")
```

```
df["building_material"] = df["building_material"].map({
    'has_superstructure_adobe_mud': 'adobe',
    'has_superstructure_mud_mortar_stone': 'stone',
    'has_superstructure_stone_flag': 'stone',
    'has_superstructure_mud_mortar_brick': 'brick',
    'has_superstructure_cement_mortar_stone': 'stone',
    'has_superstructure_cement_mortar_brick': 'brick',
    'has_superstructure_timber': 'wood',
    'has_superstructure_bamboo': 'wood'
}).fillna("other")
```

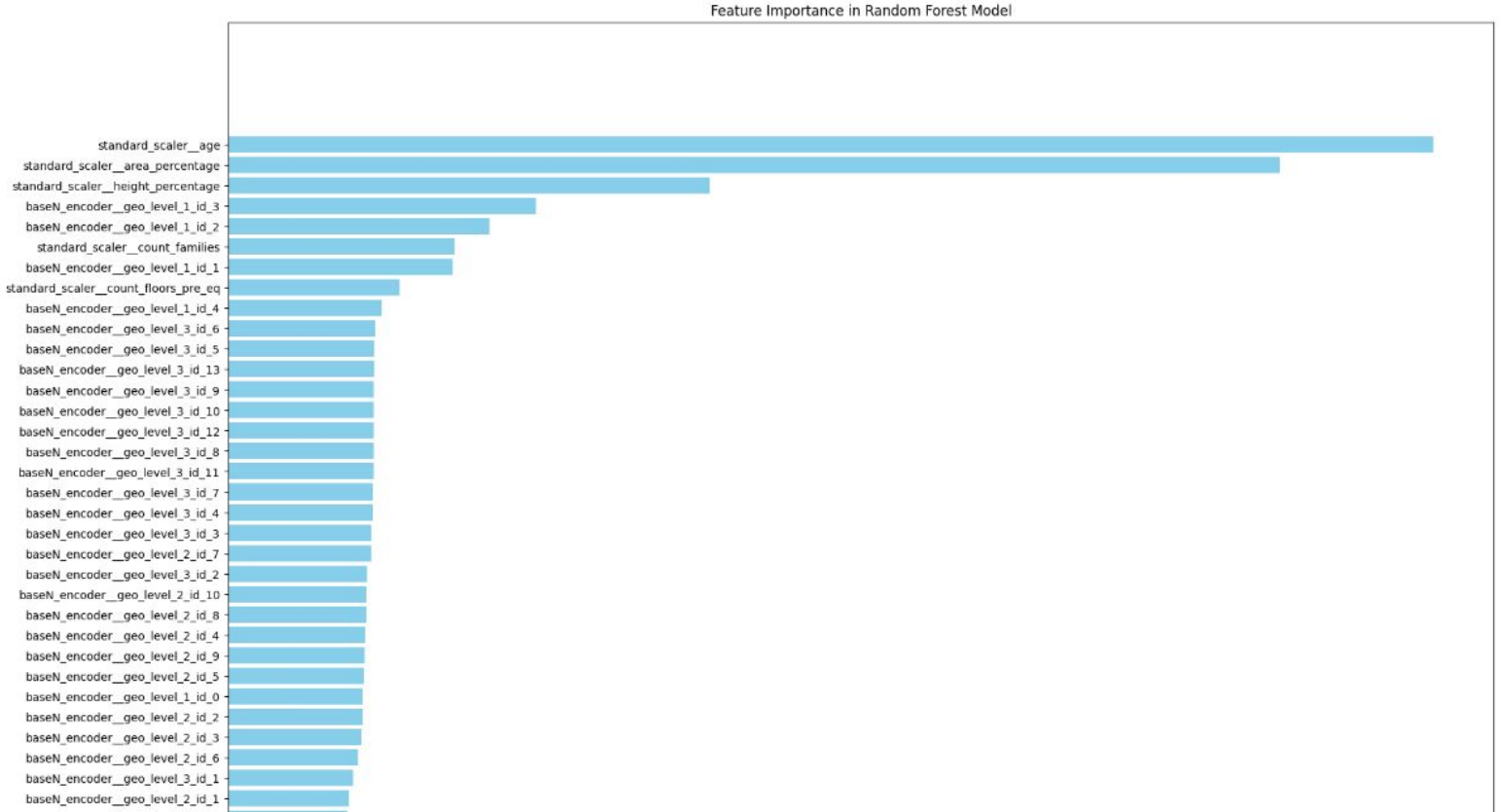
Feature engineering

```
categorical_feature
```

```
['geo_level_1_id',
 'geo_level_2_id',
 'geo_level_3_id',
 'land_surface_condition',
 'foundation_type',
 'roof_type',
 'ground_floor_type',
 'other_floor_type',
 'position',
 'plan_configuration',
 'has_superstructure_adobe_mud',
 'has_superstructure_mud_mortar_stone',
 'has_superstructure_stone_flag',
 'has_superstructure_cement_mortar_stone',
 'has_superstructure_mud_mortar_brick',
 'has_superstructure_cement_mortar_brick',
 'has_superstructure_timber',
 'has_superstructure_bamboo',
 'has_superstructure_rc_non_engineered',
 'has_superstructure_rc_engineered',
 'has_superstructure_other',
 'legal_ownership_status',
 'has_secondary_use',
 'has_secondary_use_agriculture',
 'has_secondary_use_hotel',
```

```
df["type_of_building"] = df["type_of_building"].map({
    'has_secondary_use_agriculture': "agriculture",
    'has_secondary_use_hotel': "institutional",
    'has_secondary_use_rental': "other",
    'has_secondary_use_institution': "institutional",
    'has_secondary_use_school': "institutional",
    'has_secondary_use_industry': "industrial",
    'has_secondary_use_health_post': "other",
    'has_secondary_use_gov_office': "institutional",
    'has_secondary_use_use_police': "institutional",
    'has_secondary_use_other': "other",
    'has_secondary_use': "other"
}).fillna("other")
```

Feature importance



Feature engineering

```
# Experiment :

# with different types of features :

# 1. all features (all old features )
# 2. all features + new build Features
# 3. all features + new build Features - has_flags
# 4 . all features - has_flags
```

1	0.7201188618338072
2	**0.7247151377813883**
3	**0.7247151377813883**
4	0.7201188618338072

Encoding and scaling

Experiment ID	geo_level_1_id	geo_level_2_id	geo_level_3_id
Exp-1	Target	Target	Target
Exp-2	BaseN	BaseN	BaseN
Exp-3	Frequency	Frequency	Frequency
Exp-4	BaseN	Target	Target
Exp-5	Target	BaseN	BaseN
Exp-6	Frequency	BaseN	BaseN
Exp-7	BaseN	Frequency	Target
Exp-8	Frequency	Target	BaseN

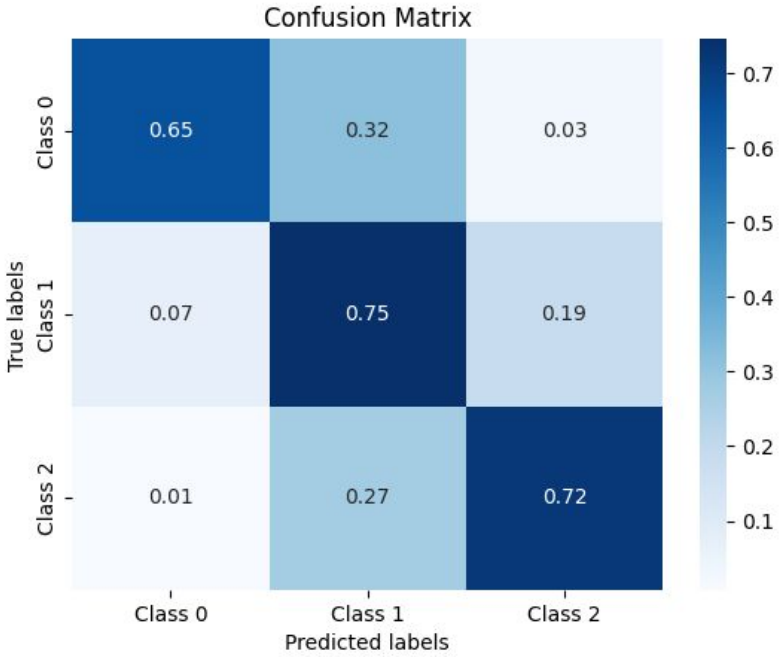
```
Testing Encoding: {'geo_level_1_id': 'target', 'geo_level_2_id': 'target', 'geo_level_3_id': 'target'}
F1 Score: 0.7283
Testing Encoding: {'geo_level_1_id': 'basen', 'geo_level_2_id': 'basen', 'geo_level_3_id': 'basen'}
F1 Score: 0.7216
Testing Encoding: {'geo_level_1_id': 'frequency', 'geo_level_2_id': 'frequency', 'geo_level_3_id': 'frequency'}
F1 Score: 0.7092
Testing Encoding: {'geo_level_1_id': 'basen', 'geo_level_2_id': 'target', 'geo_level_3_id': 'target'}
F1 Score: 0.7276
Testing Encoding: {'geo_level_1_id': 'target', 'geo_level_2_id': 'basen', 'geo_level_3_id': 'basen'}
F1 Score: 0.7221
Testing Encoding: {'geo_level_1_id': 'frequency', 'geo_level_2_id': 'basen', 'geo_level_3_id': 'basen'}
F1 Score: 0.7191
Testing Encoding: {'geo_level_1_id': 'basen', 'geo_level_2_id': 'frequency', 'geo_level_3_id': 'target'}
F1 Score: 0.7265
Testing Encoding: {'geo_level_1_id': 'frequency', 'geo_level_2_id': 'target', 'geo_level_3_id': 'basen'}
F1 Score: 0.7273
```

Encoding

```python
# Define multiple scalers for numerical features
scalers = {
    **{col: 'standard+robust' for col in num_features3},  # Apply both Standard and Robust Scaler
    **{col: 'minmax' for col in []},  # If minmax is used separately
}

# Define multiple encodings for categorical features
encoders = {
    **{col: 'basen+target' for col in ['geo_level_1_id', 'geo_level_2_id', 'geo_level_3_id']},  # Multiple encodings for geo features
    **{col: 'onehot' for col in ['plan_configuration', 'foundation_type', 'ground_floor_type',
                                 "other_floor_type", 'building_material', 'type_of_building',
                                 'position', 'legal_ownership_status', 'roof_type', 'land_surface_condition']},
    **{col: 'binary' for col in ['is_concrete', 'sticking_material']}
}
```

F1 Score: 0.7314492164952862



Modelling

- Added new models to Stacking:

```
HistGradientBoostingClassifier
KNeighborsClassifier
RidgeClassifier
QuadraticDiscriminantAnalysis
SGDClassifier
```

KNClass is repeated but with different numbers of neighbours (15)

- Increased the size of models parameters (e.g. n_estimators, iterations…)
- Added log transformed numerical variables to the features

Final Result